

**UNITED STATES PATENT APPLICATION**

of

**Alex Hopmann**

**Becky Anderson**

**and**

**Brian Deen**

for

**SYSTEMS AND METHODS FOR  
DETECTING AND RESOLVING RESOURCE CONFLICTS**

WORKMAN, NYDEGGER & SEELEY

A PROFESSIONAL CORPORATION

ATTORNEYS AT LAW

1000 EAGLE GATE TOWER

60 EAST SOUTH TEMPLE

SALT LAKE CITY, UTAH 84111

## BACKGROUND OF THE INVENTION

### **1. The Field of the Invention**

The present invention relates to systems and methods for replication and synchronization of data or resources between servers and clients. More specifically, the present invention relates to systems and methods for detecting and resolving conflicts as resources are replicated between servers and clients.

### **2. The Prior State of the Art**

Technology is increasing and expanding at an incredible rate and is permitting computers to be integrated into many different facets of people's lives. People are able to use computers to expand the confines of their own hard drive. Instead of simply performing tasks such as word processing and being limited to the information that can be locally stored, people are increasingly connected to networks which expand the amount of information that is available to them.

The information available across a network, including the Internet, can be downloaded quickly. More importantly, the data or information can be downloaded as needed. For example, many businesses have implemented a network where documents and other resources are stored on a server that is accessible by all clients instead of storing all of a business's documents and resources on each local hard drive. Clients are able to access only the resources that they require from the server. After the resource has been modified or used, it is restored on the server and is available for another client. In some instances, the same document can be accessed by more than one client simultaneously. A significant drawback of this arrangement is that the resource must be downloaded every time the client needs to work with the resource.

1 The internet, local area networks, intranets and business computer networks are  
2 becoming more sophisticated and complex which enables businesses and individuals to  
3 share information more quickly and easily. Interoffice email and calendaring applications  
4 are becoming a necessity as networks become an integral part of business. Never before has  
5 so much information been readily available and accessible by individuals and businesses.

6 This ability to access and share data has led to some rather unexpected problems.  
7 For instance, it is difficult to work on a network resource when a person or client is not  
8 connected to the network. Even if the resource is downloaded before the client goes offline,  
9 the client is not assured of working on the most recent version of the data or resource. In  
10 fact, the client may overwrite the changes of another client when the changed resource is  
11 uploaded to a server. However, the ability to access network resources while offline is a  
12 pressing need in today's age of portable computers.

13 Another problem is related to the amount of network traffic as the clients connected  
14 to a network download resources, upload resources, transmit email, read email and utilize  
15 calendar applications and use the resources of a computer network. In fact, the speed of a  
16 network can be significantly impaired by the amount of traffic or data that is being  
17 transmitted over the network. In a system having more than one server, these problems are  
18 partially solved by having server-to-server data replication to insure that each server has the  
19 most recent version of a resource. This is much more difficult to perform with clients  
20 because they are not always connected to the network and are not easily accessible by other  
21 clients.

22 Replication, however, has the potential to reduce network traffic and make clients  
23 more efficient, but there are several problems related to server to client replication. One  
24 issue to be addressed in minimizing the transfer of resources over the network while

1 ensuring that the client has the most up to date version of a resource. Another issue to be  
2 addressed is related to the conflicts which may exist between the resources stored on the  
3 clients and the resources stored on the servers.

4 Conflicts between a client copy of a resource and a server copy of a resource can  
5 occur because multiple copies of a particular resource can reside on multiple servers and  
6 multiple clients. A simple example of this conflict is where two clients have downloaded a  
7 resource from a server. The first client deletes a portion of the resource and uploads the  
8 altered resource to the server. The second client adds a different portion to the resource, but  
9 does not delete the portion deleted by the first client. The second client then uploads the  
10 changed resource to the server. In this scenario, the portion deleted by the first client is in  
11 effect restored because the version of the first client was overwritten by the changes made  
12 by the second client. Conflicts present a difficult problem.

13 One solution to this problem is to only allow a single client to download a particular  
14 resource. While this solution prevents conflicts between multiple copies of a resource, the  
15 ability to share data and work collaboratively with a resource is severely hampered.  
16 Another solution to the conflict problem is to initiate an arbitration process between the  
17 entities involved in the conflict. While this solution may resolve the conflict, the process  
18 produces a significant amount of network traffic and may require the interaction of an end  
19 user, both of which lower the efficiency of the network and the users. This problem is  
20 magnified when multiple servers and clients are involved in the arbitration process.

21 Because of the eventuality of a conflict between the copies of resources on clients  
22 and the copies of resources on the servers, it is desirable and would be an advance in the art  
23 to be able to detect and resolve conflicts. It would also be an advance in the art to minimize  
24 the amount of network traffic needed to resolve the conflict.

**WORKMAN, NYDEGGER & SEELEY**  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

$$Y$$

**AN, NYDEGGER & S**  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

When a client downloads a resource, the resource tag is also transmitted to the client and the client keeps track of the resource tag. After the client has made changes to the resource and desires to upload the changes to the server, the resource tag is transmitted to the server. The server compares the value of the resource tag sent by the client with the value of the resource tag maintained by the server and if the resource tags match, then the client is uploading changes to the version of the resource that was downloaded. If the

1 resource tags do not match, then a change was made to the resource that the client is  
2 unaware of and a conflict is thereby detected.

3       Once a conflict has been detected, it must be resolved. An exemplary method used  
4 to resolve the conflict has three stages or levels, each of which may be independent of the  
5 others. The first level of conflict resolution occurs at the server. The server recognizes that  
6 a conflict exists, by comparing the resource tags, and may evaluate the conflicting resources  
7 to determine if the conflict can be resolved by the server. Resolving the conflict at the  
8 server provides the advantage of reducing network traffic as well as relieving an end user of  
9 the task of resolving the conflict.

10       If the conflict cannot be resolved by the server, the second level of conflict resolution  
11 is initiated. Preferably, the second level of conflict resolution occurs at the client. The  
12 client is notified by the server that the conflict could not be resolved at the server and the  
13 server provides the client with information about the conflicting copies of the resource. This  
14 information, or conflict resource, may be limited to only the parts of the resources that were  
15 changed or may be as expansive as to include a complete copy of each conflicting copy of  
16 the resource. The client attempts to resolve the conflict by, for example, incorporating  
17 additional information about the operation that created the conflict or by using knowledge of  
18 the scope of operations that are known to the client, but not to the server. In general, the  
19 client typically has more ability to resolve a conflict because a client is familiar with the  
20 schema within which the resource was changed. Knowledge of this schema may assist the  
21 client in resolving the conflict. If the client cannot resolve the conflict, then an end user is  
22 involved who typically resolves the conflict.

23       Once a conflict has been resolved, the resource is stored on the server, and a new  
24 resource tag is generated, if necessary. The resource tag is provided to the clients in cases

1 where this is required. In this manner detecting and resolving resource conflicts minimizes  
2 the amount of network bandwidth needed to resolve the conflict. Also the intervention of an  
3 end user is preferably used as a final remedy.

4 Additional objects and advantages of the invention will be set forth in the description  
5 which follows, and in part will be obvious from the description, or may be learned by the  
6 practice of the invention. The objects and advantages of the invention may be realized and  
7 obtained by means of the instruments and combinations particularly pointed out in the  
8 appended claims. These and other objects and features of the present invention will become  
9 more fully apparent from the following description and appended claims, or may be learned  
10 by the practice of the invention as set forth hereinafter.

**WORKMAN, NYDEGGER & SEELEY**  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

Figure 4 is a flow chart of one embodiment for detecting and resolving resource conflicts.



**DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

When data is replicated from server to client and from client to server, there is always a possibility that a conflict may exist between the client copy of the resource and the server copy of the resource. Detecting and resolving resource conflicts is important in order to assure resource integrity. Without conflict detection and resolution, a client cannot be assured of the status of a resource stored on a server. Conflict detection and resolution allows a client to upload a resource to a server without the fear of overwriting the changes made to the resource by another client.

Conflict resolution can occur at the server, the client or with an end user. Preferably the resolution occurs at the server, in part because the conflicting resources need not be transmitted over the network. Also, the end user need not be involved in resolving the conflict. Thus, conflict resolution makes a network more efficient and relieves an end user of having to resolve some conflicts. Conflict resolution can also occur at a client, but the client preferably attempts to resolve the conflict only if the server is unable to fix the conflict. Finally, an end user becomes involved in the conflict resolution only if both the server and the client are unable to resolve the conflict.

The invention is described below by using diagrams to illustrate either the structure or processing of embodiments used to implement the systems and methods of the present invention. Using the diagrams in this manner to present the invention should not be construed as limiting of its scope. The present invention contemplates both methods and systems for detecting and resolving resource conflicts. The embodiments of the present invention may comprise a special purpose or general-purpose computer including various computer hardware, as discussed in greater detail below.

1           Embodiments within the scope of the present invention also include computer-  
2 readable media for carrying or having computer-executable instructions or data structures  
3 stored thereon. Such computer-readable media can be any available media which can be  
4 accessed by a general purpose or special purpose computer. By way of example, and not  
5 limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM  
6 or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any  
7 other medium which can be used to carry or store desired program code means in the form  
8 of computer-executable instructions or data structures and which can be accessed by a  
9 general purpose or special purpose computer. When information is transferred or provided  
10 over a network or another communications connection (either hardwired, wireless, or a  
11 combination of hardwired or wireless) to a computer, the computer properly views the  
12 connection as a computer-readable medium. Thus, any such a connection is properly termed  
13 a computer-readable medium. Combinations of the above should also be included within  
14 the scope of computer-readable media. Computer-executable instructions comprise, for  
15 example, instructions and data which cause a general purpose computer, special purpose  
16 computer, or special purpose processing device to perform a certain function or group of  
17 functions.

18           Figure 1 and the following discussion are intended to provide a brief, general  
19 description of a suitable computing environment in which the invention may be  
20 implemented. Although not required, the invention will be described in the general context  
21 of computer-executable instructions, such as program modules, being executed by  
22 computers in network environments. Generally, program modules include routines,  
23 programs, objects, components, data structures, etc. that perform particular tasks or  
24 implement particular abstract data types. Computer-executable instructions, associated data

1 structures, and program modules represent examples of the program code means for  
2 executing steps of the methods disclosed herein. The particular sequence of such executable  
3 instructions or associated data structures represent examples of corresponding acts for  
4 implementing the functions described in such steps.

5 Those skilled in the art will appreciate that the invention may be practiced in  
6 network computing environments with many types of computer system configurations,  
7 including personal computers, hand-held devices, multi-processor systems, microprocessor-  
8 based or programmable consumer electronics, network PCs, minicomputers, mainframe  
9 computers, and the like. The invention may also be practiced in distributed computing  
10 environments where tasks are performed by local and remote processing devices that are  
11 linked (either by hardwired links, wireless links, or by a combination of hardwired or  
12 wireless links) through a communications network. In a distributed computing environment,  
13 program modules may be located in both local and remote memory storage devices.

14 With reference to Figure 1, an exemplary system for implementing the invention  
15 includes a general purpose-computing device in the form of a conventional computer 20,  
16 including a processing unit 21, a system memory 22, and a system bus 23 that couples  
17 various system components including the system memory 22 to the processing unit 21. The  
18 system bus 23 may be any of several types of bus structures including a memory bus or  
19 memory controller, a peripheral bus, and a local bus using any of a variety of bus  
20 architectures. The system memory includes read only memory (ROM) 24 and random  
21 access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic  
22 routines that help transfer information between elements within the computer 20, such as  
23 during start-up, may be stored in ROM 24.

The computer 20 may also include a magnetic hard disk drive 27 for reading from and writing to a magnetic hard disk 39, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to removable optical disk 31 such as a CD-ROM or other optical media. The magnetic hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive-interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer-executable instructions, data structures, program modules and other data for the computer 20. Although the exemplary environment described herein employs a magnetic hard disk 39, a removable magnetic disk 29 and a removable optical disk 31, other types of computer readable media for storing data can be used, including magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, RAMs, ROMs, and the like.

Program code means comprising one or more program modules may be stored on the hard disk 39, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may enter commands and information into the computer 20 through keyboard 40, pointing device 42, or other input devices (not shown), such as a microphone, joy stick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 coupled to system bus 23. Alternatively, the input devices may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor 47 or another display device is also connected to system bus 23 via an interface, such as video adapter 48.

1 In addition to the monitor, personal computers typically include other peripheral output  
2 devices (not shown), such as speakers and printers.

3 The computer 20 may operate in a networked environment using logical connections  
4 to one or more remote computers, such as remote computers 49a and 49b. Remote  
5 computers 49a and 49b may each be another personal computer, a server, a router, a network  
6 PC, a peer device or other common network node, and typically includes many or all of the  
7 elements described above relative to the computer 20, although only memory storage  
8 devices 50a and 50b and their associated application programs 36a and 36b have been  
9 illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area  
10 network (LAN) 51 and a wide area network (WAN) 52 that are presented here by way of  
11 example and not limitation. Such networking environments are commonplace in office-  
12 wide or enterprise-wide computer networks, intranets and the Internet.

13 When used in a LAN networking environment, the computer 20 is connected to the  
14 local network 51 through a network interface or adapter 53. When used in a WAN  
15 networking environment, the computer 20 may include a modem 54, a wireless link, or other  
16 means for establishing communications over the wide area network 52, such as the Internet.  
17 The modem 54, which may be internal or external, is connected to the system bus 23 via the  
18 serial port interface 46. In a networked environment, program modules depicted relative to  
19 the computer 20, or portions thereof, may be stored in the remote memory storage device. It  
20 will be appreciated that the network connections shown are exemplary and other means of  
21 establishing communications over wide area network 52 may be used.

22 As used herein, server refers to computer hardware and software that manage  
23 network resources. Client, as used herein is an application or software that is executing on a  
24 computer that relies on the server to perform some operations. Download refers to the

1 transfer of data from the server to the client and upload refers to the transfer of data from the  
2 client to the server. In some instances a read operation is equivalent to a download and a  
3 write operation is equivalent to an upload.

4 Figure 2 is a block diagram illustrating an exemplary network environment in which  
5 the systems and methods of the present invention may operate. Network 20 comprises a  
6 plurality of servers and clients, each of which has resources. Resources, as used herein,  
7 refers to objects in a namespace that may be referred to by a Uniform Resource Identifier  
8 (URI) and include, but are not limited to, files, folders, directories, collections, documents,  
9 spreadsheets, emails, messages and others. A collection, in one embodiment, is a group of  
10 resources.

11 An important aspect of the plurality of servers in network 20 is that they may be  
12 versioning servers, which is capable of tracking and storing different versions of a particular  
13 resource. When a change is made to a resource, the current and the old versions of the  
14 resource are tracked. The methods and systems of the present invention may be applied to  
15 versioning servers as well as other types of servers.

16 Resource 28 in Figure 2 illustrates one embodiment of a resource. Resource 28 has  
17 content 29 and properties 30. If resource 28 is an email, then content 29 corresponds to the  
18 actual text of the message and properties 30 may refer to a variety of items which may  
19 include the size of the email, the sender of the email, and the importance of the email. There  
20 are many more properties which can be associated with a resource that are not detailed  
21 herein.

22 In a preferred embodiment, the resources of a network are accessed and replicated  
23 using WebDAV methods and XML elements. The methods implemented by embodiments  
24 of the present invention are similar to the methods defined in HTTP with the difference that

1 replication is supported. These methods and elements will be discussed as necessary with  
2 respect to various embodiments of the present invention. An important aspect of the  
3 WebDAV methods and XML elements is that they provide support for conflict detection and  
4 conflict resolution. For instance, the GET method allows a client to include a resource tag  
5 such that the resource is downloaded only if the server copy of the resource is different from  
6 the copy of the resource on the client. In the alternative, the GET method allows a client to  
7 download a particular version of a resource by identifying the particular version with the  
8 resource tag.

9       The resources of the servers in network 20 may or may not be identical. In many  
10 instances resources are replicated across servers and each client can therefore access a  
11 particular resource from one or more servers. The resources that are the same are typically  
12 replicated between the servers such that the most current version of a particular resource is  
13 available to the clients. Additionally, a client has the ability to synchronize or replicate its  
14 resources with the resources of the servers. One objective of synchronization or replication  
15 is to avoid downloading data that has already been downloaded by a client. As described  
16 previously, a GET method may include a resource tag which may be used to determine in a  
17 download of the contents of the resource is necessary. This reduces network data traffic and  
18 enables clients to be more efficient when they are connected to the network as well as when  
19 they are not connected to the network because they may work on the resources even when  
20 not connected to network 20.

21       For example, when client 25 connects to network 20 for the first time, resources 22  
22 and resources 24 are not physically present on client 25. Client 25 has the ability to  
23 download resources from either server 21 or server 23. Client 25 may download all or part  
24 of resources 22 and resources 24. In addition to the resources downloaded by client 25,

a  
1 client 25 receives an identifier which is indicative of state of the resources that were  
2 requested and downloaded by client 25. The identifier is a binary stream that represents the  
3 state of the contents of a collection and is more fully described in co-pending U.S. patent  
4 application serial number 09/412,739 entitled "METHOD, COMPUTER  
5 READABLE MEDIUM, AND SYSTEM FOR MONITORING THE STATE OF A  
6 COLLECTION OF RESOURCES", filed on October 4, 1999, which assigned to the same  
7 assignee as the present application, and which is hereby incorporated by reference. Also,  
8 the client receives a resource tag for each resource.

9 After client 25 has downloaded resources 26, client 25 may disconnect from the  
10 network and use resources 26, which were downloaded from servers 21 and/or 23. When  
11 client 25 reconnects to network 20 at a later time, client 25 can send its identifier to server  
12 21 or 23 and the server will make a determination of what has changed since client 25  
13 downloaded by comparing the identifier of the client with the current identifier maintained  
14 by the server. After the server has compared the identifier sent by the client with the current  
15 identifier of the collection, the server sends client 25 the current identifier and a manifest  
16 indicating the resources that have changed. The manifest may include the resource tags of  
17 the resources that the client has not yet downloaded. This allows client 25 to download only  
18 those resources that have changed using the GET method rather than download all the  
19 resources again. Thus, a client is relieved of having to repeatedly download resources and  
20 network traffic is reduced. In a preferred embodiment, a client will only have to download a  
21 resource one time.

22 In this manner, the resources are available for use by client 25 as if client 25 were  
23 connected to network 20 at all times. The systems and methods needed to replicate,  
24 synchronize and store the resources between servers and clients is more fully described in



WORKMAN, NYDEGGER & SEELEY  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

co-pending U.S. patent application serial no <sup>09/412,766</sup> entitled "Method and System for Supporting Off-Line Mode of Operation and Synchronization", filed on October 4, 1999, which assigned to the same assignee as the present application, and which is hereby incorporated by reference. In one embodiment, replication is a process initiated by a client of retrieving a manifest from a server, downloading the resources that have been changed by other clients and uploading the resources changed by the client to the server. In this manner, the client has a current and complete local copy of the resources on the server.

As previously described, one objective of the present invention is to only download a resource, a group of resources, or a collection once and the identifier enables the client to meet this objective which has the effect of reducing network traffic and permitting the client to work with up to date information even if the client is not connected to the network or is offline. The resources downloaded from the servers are cached by the client such that the client has offline access to resources. When the client reconnects with network 20 or is again online, replication or synchronization as described previously may occur.

The identifier is generated and maintained by the server. The client, in a preferred embodiment, is unable to interpret the contents of the identifier, but simply stores the identifier as indicative of the status of a collection or resource located on a server. Another identifier generated by a server is a resource tag which indicates the status or state of a particular resource. In other words, the resource tag identifies a particular version of a particular resource. When a client seeks a new identifier and manifest from a server, the resource tags of the resources that have changed are returned to the client in the manifest.

A resource tag cannot be understood or generated by a client, but a client is capable of comparing two or more resource tags. The comparison preferably performed determines whether the resource tags are identical or have exact equality. In a preferred embodiment,

1 the client stores resource tags in order to reflect the state of the replicated resource. The  
2 resource tag is therefore useful and is in fact preferred if a client desires to detect and resolve  
3 resource conflicts. Resource tags are more fully described in copending U.S. patent  
4 application serial number 09/412,071, entitled "METHOD, DATA STRUCTURE, AND  
5 COMPUTER PROGRAM PRODUCT FOR IDENTIFYING A NETWORK RESOURCE",  
6 filed on October 4, 1999, which assigned to the same assignee as the present application,  
7 and which is hereby incorporated by reference.

8 Because a conflict occurs when there is a difference between the server copy of the  
9 resource and the client copy of a resource, resource tags are particularly useful for detecting  
10 and resolving conflicts. Figure 3 is a block diagram illustrating one situation where a  
11 conflict may occur and is illustrative of server side conflict detection. Server 40 of Figure 3  
12 has resource 41 and server 40 has generated resource tag 42 which is a property of resource  
13 41 and identifies a particular version of resource 41.

14 In this embodiment, the GET method is used to retrieve resource 41 from server 40  
15 by clients 43 and 44. In order to provide conflict detection, server 40 must also provide  
16 resource tag 42 to each client requesting to download resource 41. Thus, both client 43 and  
17 client 44 each have a copy of resource 41 and a copy of resource tag 42. Next, client 43  
18 alters resource 41 and uses the PUT method to update the existing resource 41 on server 40.  
19 In a PUT method, the client's saved resource tag may be sent in an IF header. Server 40  
20 compares the resource tag provided by client 43 in the IF header with the resource tag  
21 maintained by the server and if the resource tag provided by client 43 matches the resource  
22 tag stored on server 40, then the PUT is allowed. Because resource 41 has changed, a new  
23 resource tag 46 is generated by server 40 and returned to client 43 to indicate the version of  
24

1 resource 41 stored on client 43. Thus both client 43 and server 40 have identical versions of  
2 resource 41 as indicated by the respective values of the resource tags.

3 In this example, client 44 attempts to update resource 41 using a PUT method with  
4 an if header containing the client's saved copy of resource tag 42. In this case, client 43 has  
5 already altered resource 41 and resource tag 42 provided by client 44 does not match the  
6 current resource tag 46 of resource 41. Because the resource tags do not match, server 40  
7 detects a conflict and client 44 is not allowed to PUT the resource until the conflict is  
8 resolved. Conflicts may be created in other situations, and the conflict is typically detected  
9 by comparing resource tags.

10 After a conflict has been detected, the next step is to resolve the conflict and Figure 4  
11 is a flowchart illustrating an exemplary method for resolving a conflict. In step 60, the  
12 conflict is detected. Server side detection 61, as previously described, is one method for  
13 detecting a conflict. Another method for detecting a conflict is client initiated conflict  
14 detection 62, which occurs when a resource downloaded by a first client is updated online  
15 by a second client while the first client was offline.

16 In this instance, the first client downloads the resource and its resource tag using, for  
17 example, a web browser. When the client then replicates or synchronizes with the server,  
18 the client requests an updated identifier and a new manifest by transmitting the identifier  
19 currently saved and stored by the client in the communication sent to the server. Because  
20 the resource in question has changed, the server returns the new resource tag to the client in  
21 the manifest. The client then compares the resource tag in the manifest with the resource tag  
22 which was downloaded using the web browser. If the resource tags match, then the client  
23 need not perform a download on the changed resource because the most recent version is  
24 already present on the client. As described earlier, while the resource tag is not understood

1 by the client, the client does have the capability of comparing resource tags. This ability  
2 enables a client to perform client side conflict detection. This is an example of client  
3 initiated conflict detection.

4 Step 64 involves the methods and processes of resolving the detected conflict.  
5 Objectives of conflict detection and resolution include minimizing the amount of network  
6 traffic, limiting the necessity of having an end user resolve the conflict, and increasing  
7 efficiently. Each sub-step of conflict resolution step 64 is independent of the other sub-  
8 steps, but the sub-steps are preferably executed in a prescribed order. Thus, the first sub-  
9 step to be executed is step 65, which is a server level conflict resolution step where the  
10 server attempts to resolve the conflict.

11 Because the server has detected the conflict, the server has an opportunity to resolve  
12 the conflict without having the client or the end user involved in the conflict resolution. By  
13 resolving the conflict at the server, the resource does not necessarily need to be downloaded  
14 to the client, which meets one of the objectives. If the copy of the resource stored by the  
15 client is not the copy of the resource that is stored on the server after the conflict is resolved  
16 by the server, then the client will have to download the resource again.

17 In a preferred embodiment, the server attempts to resolve the conflict by comparing  
18 the client version of the resource with the server version of the resource. In other words, the  
19 server is comparing the changes requested by the client with the existing state of the  
20 resource. If the server determines that the changes requested by the client and the existing  
21 state of the resource are semantically equivalent, then the conflict may be resolved at the  
22 server level. For instance, if two clients both changed a property, such as the author  
23 property, to the same person, then when the second client attempts to upload the resource  
24 properties, the resource tags will not match. The server, however, can determine that there



1 For instance, if a meeting is scheduled in a calendar application and the first client  
2 dismisses the meeting at a certain time and the second client dismisses the meeting at a later  
3 time, a conflict resource is generated because the second client is attempting to set the  
4 dismiss property which was already altered by the first client. The client, which has  
5 knowledge of how the calendar application operates, is able to resolve this conflict without  
6 involving the end user. The client can recognize that the change of each client had the effect  
7 of dismissing the meeting. The client can resolve the conflict by leaving the server copy of  
8 the resource unaltered, and simply updating its own local copy of the resource tag with the  
9 server's current resource tag. This avoids any unnecessary or redundant changes to the  
10 server data, as well as changes to the server's resource tag, which would cause all other  
11 clients to identify their copies of the resource as old and cause the clients to download the  
12 resource again. Resolving the conflict at the client is an example of client level conflict  
13 resolution.

14 In some situations, it might be possible for the server to have similar knowledge of  
15 various schemas, allowing the server to recognize two changes as being identical. If is  
16 preferable that the server resolve resource conflicts, resulting again in reduced network  
17 traffic without requiring creation of a conflict resource, transmission of a conflict resource,  
18 or intelligence in the client to correctly detect that two changes are equivalent. If the server  
19 cannot resolve the conflict, it is preferable that the client resolve the conflict.

20 If the client is unable to resolve the conflict, then the conflict resource is presented to  
21 the end user for resolution in step 68. The user is presented with the different versions of the  
22 resource as well as the conflict and the user is able to remedy the conflict. After the conflict  
23 is resolved in step 68, the resolved resource is uploaded to the server and a new resource tag  
24 is returned. In sum, conflict resolution can be viewed as occurring in stages or levels. The

1 first level involves resolution at the server, the second level of conflict resolution occurs at  
2 the client and the third level of conflict resolution is the interaction with the end user and as  
3 previously mentioned, it is preferable for the server to resolve the resource conflict.

4 The present invention may be embodied in other specific forms without departing  
5 from its spirit or essential characteristics. The described embodiments are to be considered  
6 in all respects only as illustrative and not restrictive. The scope of the invention is,  
7 therefore, indicated by the appended claims rather than by the foregoing description. All  
8 changes which come within the meaning and range of equivalency of the claims are to be  
9 embraced within their scope.

10 What is claimed and desired to be secured by United States Letters Patent is:  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24